ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4
ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ

Дисциплина: CSE842 «Capstone Project». **Образовательная программа:** 6B06102 «Computer Science», 4 курс, 8 семестр. **Связь с лекцией:** лекция 4 «Архитектурный синтез программной системы». **Результаты обучения:** РО3, РО4. **Объём:** 3 часа практики, 7 часов самостоятельной работы, включая СРСП. **Максимальный балл:** 3.

Цель занятия: спроектировать архитектуру системы: выбрать драйверы и стиль, выделить подсистемы и потоки данных, зафиксировать решения в ADR.

Задачи:

1. Выбрать архитектурные драйверы и сформулировать 3–4 сценария атрибутов качества.
2. Выбрать архитектурный стиль и выделить подсистемы с ответственностью и владельцами данных.
3. Построить диаграммы C4 (контекст, контейнеры, компоненты ключевого контейнера) и схему потоков данных.
4. Оформить не менее трёх ADR.

Программное обеспечение и инструменты: draw.io, PlantUML C4, Structurizr DSL или Mermaid; Markdown.

1. Краткие теоретические сведения

Архитектурные драйверы — требования, сильнее всего влияющие на структуру: ключевые FR, приоритетные атрибуты качества, ограничения. Сценарий атрибута качества состоит из источника стимула, стимула, артефакта, среды, отклика и меры отклика. Модель C4 включает уровни контекста, контейнеров, компонентов и кода. ADR фиксирует контекст, альтернативы, решение и последствия.

Пример описания контейнеров в Mermaid:

```
flowchart LR
  U([Пользователь]) -->|HTTPS| W[Веб-клиент]
  W -->|REST JSON| A[API-сервис]
  A -->|SQL| D[(БД)]
  A -->|событие| Q[[Очередь]]
  Q --> N[Уведомления]
```

Таблица 1 — Шаблон ADR

Поле	Содержание
Номер и название	ADR-0003. Уведомления отправляются асинхронно
Контекст	почтовый сервис отвечает 1–5 с; драйверы NFR-1 и доступность
Альтернативы	синхронная отправка; таблица outbox; брокер RabbitMQ
Решение	outbox и фоновый обработчик

Поле	Содержание
Последствия	ответ API не зависит от почты; задержка письма до 1 мин
Статус	принято

2. Порядок выполнения работы

1. Выпишите из SRS 5–7 архитектурных драйверов и сформулируйте сценарии качества.
2. Рассмотрите 2–3 архитектурных стиля и выберите один с обоснованием.
3. Выделите подсистемы (таблица 2 — минимальный набор), опишите их ответственность и интерфейсы.
4. Постройте диаграммы C4 как код и сохраните в docs/architecture/.
5. Нарисуйте поток данных сквозного сценария с отметкой синхронных и асинхронных связей.
6. Подберите тактики для каждого сценария качества и проверьте архитектуру сценариями.
7. Оформите ADR (не менее трёх), включая решение по вопросу варианта.

3. Индивидуальные задания

Номер варианта совпадает с номером темы проекта, выбранной на практическом занятии № 1, и сохраняется до конца семестра. Команда (до 4 человек) выполняет задание своего варианта; распределение работы между участниками указывается в отчёте.

Таблица 2 — Индивидуальные задания: подсистемы и драйвер

№	Подсистемы (минимум)	Главный драйвер	Тема обязательного ADR
1	веб-клиент, API записи, БД расписаний, SMS-уведомления	функциональная корректность	хранение и API: Django + PostgreSQL; Node.js + MongoDB; Firebase
2	мобильное приложение, генератор QR, API отметок, БД, отчёты	безопасность (подлинность)	клиентское приложение: Flutter; React Native; PWA
3	датчик ESP32 (или эмулятор), MQTT-брокер, сервис приёма, БД временных рядов, дашборд	надёжность	хранилище временных рядов: InfluxDB; TimescaleDB; SQLite

№	Подсистемы (минимум)	Главный драйвер	Тема обязательного ADR
4	веб-интерфейс, API, очередь задач, песочница исполнения, БД результатов	безопасность	песочница: Docker; firejail; Judge0
5	каталог, корзина и заказ, рекомендательный сервис, платёжная заглушка	функциональная корректность	рекомендации: популярные товары; item-kNN; ALS
6	каталог с поиском, выдача и возврат, уведомления о сроках, сканер штрихкодов	функциональная пригодность	полнотекстовый поиск: PostgreSQL FTS; Elasticsearch; SQLite FTS5
7	Telegram-бот, сервис поиска по базе знаний, админ-панель, журнал обращений	функциональная корректность	поиск ответа: TF-IDF; BM25; векторные эмбеддинги
8	камера (видеофайлы), сервис распознавания занятости, API, веб-карта	функциональная корректность	распознавание занятости: классический CV; CNN; ультразвуковые датчики
9	мобильный сканер, API, БД остатков, отчёты	целостность	офлайн-работа: SQLite + синхронизация; PouchDB/CouchDB; Firebase offline
10	приложение клиента, приложение курьера (GPS), диспетчер заказов, карта	производительность	передача координат: WebSocket; Server-Sent Events; периодический опрос
11	сборщик журналов, парсер, детектор аномалий, оповещения, дашборд	функциональная корректность	детектор: правила; Isolation Forest; LSTM-автоэнкодер
12	мобильное приложение, распознавание чека (OCR), API синхронизации, аналитика	функциональная корректность	распознавание: Tesseract; EasyOCR; облачный OCR

№	Подсистемы (минимум)	Главный драйвер	Тема обязательного ADR
13	портал заявлений, маршрут согласования, генерация PDF, подпись, архив	безопасность (целостность)	генерация PDF: WeasyPrint; LaTeX; DOCX-шаблон с конвертацией
14	веб-клиент, API, БД, оплата (заглушка), уведомления с QR-пропуском	целостность	блокировка слота: блокировка в БД; распределённая блокировка Redis; оптимистическая версия
15	устройства (ESP32 или эмулятор), MQTT-брокер, сервер правил, мобильное приложение	надёжность	брокер: Mosquitto; EMQX; HTTP-опрос
16	камера (видеофайлы), сервис распознавания номеров, БД пропусков, шлагбаум (эмулятор), журнал	функциональная корректность	распознавание: EasyOCR; детектор YOLO + OCR; OpenALPR
17	счётчики (эмулятор Modbus), сборщик данных, БД, прогноз, дашборд	функциональная корректность	прогноз: сезонная наивная модель; градиентный бустинг; LSTM
18	банк вопросов, проведение теста с таймером, автопроверка, аналитика	надёжность	серверный фреймворк: Django; FastAPI; Node.js Express
19	загрузчик GTFS, планировщик маршрутов, API, веб-карта	функциональная корректность	алгоритм маршрутизации: Дейкстра по графу; Connection Scan Algorithm; RAPTOR
20	календарь консультаций, синхронизация с Google Calendar, видеоссылки, напоминания	функциональная корректность	синхронизация календаря: Google Calendar API; CalDAV; экспорт iCal

4. Содержание отчёта

1. Таблица драйверов и сценарии качества.
2. Диаграммы C4 (контекст, контейнеры, компоненты) с подписями технологий и протоколов.
3. Схема потоков данных сквозного сценария.
4. Не менее трёх ADR.
5. Таблица проверки архитектуры сценариями качества.

5. Форма сдачи

Артефакты занятия размещаются в репозитории проекта (каталог docs/ или указанный в задании), в трекере задач создаются и закрываются соответствующие задачи. Преподавателю сдаётся ссылка на версию (коммит или тег) и краткий отчёт.

6. Критерии оценивания

Максимальный балл за работу — **3** (календарный план силлабуса). Баллы распределяются по трём критериям силлабуса.

Таблица 3 — Критерии оценивания текущего задания

Критерий	Доля	Что проверяется
Полнота и соответствие требованиям задания	40%	все пункты задания и индивидуального варианта выполнены
Техническая корректность	40%	решения обоснованы, расчёты и код верны, терминология точна
Проверяемость материалов	20%	артефакты в репозитории, ссылки, команды запуска, данные

Работа сдаётся через канал, указанный преподавателем; в командном проекте каждый участник указывает свой вклад. Использование генеративного ИИ раскрывается по правилам университета.

7. Контрольные вопросы

1. Что такое архитектурный драйвер?
2. Из каких частей состоит сценарий атрибута качества?
3. Почему для вашего проекта выбран именно этот стиль?
4. Какие уровни C4 вы построили?
5. Что содержит ADR?

Литература

1. Bass L., Clements P., Kazman R. Software Architecture in Practice. — 4th ed. — Addison-Wesley, 2021.
2. Richards M., Ford N. Fundamentals of Software Architecture. — 2nd ed. — O’Reilly, 2025.

3. Brown S. The C4 model for visualising software architecture. — c4model.com.